UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,540	01/12/2001	Subramanian Srinivasan	CIS00-3839	1172

7590 04/11/2005

Barry W. Chapin, Esq.
Chapin & Huang, L.L.C.
Westborough Office Park
1700 West Park Drive
Westborough, MA 01581

EXAMINER	
JARRETT, SCOTT L	

ART UNIT	PAPER NUMBER
3623	

DATE MAILED: 04/11/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/759,540	SRINIVASAN, SUBRAMANIAN	
	Examiner	Art Unit	
	Scott L. Jarrett	3623	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 February 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-34* is/are pending in the application.

4a) Of the above claim(s) *13, 28* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12,14-27 and 29-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This **Final** Office Action is responsive to Applicant's amendment filed February 22, 2005. Applicant's amendment of February 22, 2005 amended the specification, amended claims 1-12, 14-27 and 29-34 and canceled claims 13 and 28. Currently claims 1-12, 14-27 and 29-34 are pending.

Response to Amendment

2. The objection to the specification in the First Office Action is withdrawn in response to the Applicant's amendment to the abstract.

3. The 35 U.S.C. ∋ 112 (2) rejections of Claims 1-10, 15-25, 30 and 33 in the First Office Action are withdrawn is response to the Applicant's amendments to Claims 1-10, 15-25, 30 and 33.

4. The 35 U.S.C. ∋ 101 rejections of Claims 1-34 in the First Office Action are withdrawn in response to the Applicant's amendments.

Response to Arguments

5. Applicant's arguments filed February 22, 2005 have been fully considered but they are not persuasive. In the Applicant's remarks, Applicant argues that neither QAD, Inc. or Orr et al. teach the step of determining if any other attributes related to the (change) order are changed based on the change order (requested changes), and if any of the attributes are changed (different), then adding (recording, supplementing) the change order (result) to indicate the differences between those attributes or the invocation of comparison logic in order to generate a change order result indicating the differences as is now claimed (Page 18).

QAD, Inc. teaches a method and system for processing the changes to orders of items offered for sale (Quad.com: Application Data Sheets Collaboration Order Management, Sales & Distribution; Pages 1 and 3-4; "...automates and streamlines the customer buying process..."; Page 3, Paragraph 1). More generally QAD teaches a comprehensive and robust suite of products that enable e-business management (business-to-business, business-consumer, MFG/PRO, order management, etc.; QAD.com: Pages 1-4; Collaborative Applications Power B2B Transactions: "Buy-Side", "Sell-Side", "What to look for in Internet Order Management"; Page 1, Paragraph 3; Pages 4-5).

QAD, Inc. is silent on the specific mechanisms utilized by its order management system (product).

Orr et al. teaches in detail the change order methods and mechanisms utilized by the method and system for processing changes to orders of items offered for sale (manufactured items being inherently produced so as to be offered for sale; Abstract; Column 1, Lines 56-68; Column 2, Lines 1-58; Figure 3) including but not limited to the step of determining if any other attributes (affected items, engineering change, manufacturing engineering change) related to the change order are changed based on the requested/desired change to the original order (master item, design), and if any of the attributes (designs being composite objects consisting of one of more items/elements) are changed (different, affected), then adding (recording, supplementing) the change order (result, modified design) to indicate the differences between those attributes or the invocation of comparison logic in order to generate a change order result indicating the differences (Column 1, Lines 56-68; Column 2, Lines 1-68; Figures 5, and 7).

Orr et al. more specifically teach that the method and system for analyzing, storing and communicating change orders wherein:

- an analysis (comparison of an change order request to existing order/design) of change orders to determine what items of the design are affected (affected items (differences, new values, attributes, etc.; engineering change object (EC); "The EC object contains administrative information about an engineering change, such as the designer making the change and the reasons for the change."; Column 2, Lines 14-17; Column 5, Lines 10-29);

- the receipt, analysis and subsequent recording/tracking of those differences (e.g. tracking and storing previous versions of the order, affected items; "Action code column indicates the reason a master item is being changed. Typical reasons include new part introduction, changing an existing part...", Column 5, Lines 13-16; Column 7, Lines 26-61; Figures 2 and 5-8d); and

- the triggering of additional logic based on those differences ("The processing logic of the Engineering Control Manager is activated when a user selects an affected item..." Column 7, Lines 28-30; Engineering Change Control Manager, Figure 1, Element 35; Column 2, Lines 2-38; Column 7, Lines 26-40).

Further it is noted that a change order minimally consists of: an original (existing) order (sale, purchase, design, etc.), a request to change the existing order and the comparison (review, analysis) of the existing order to requested changes (change order) in order to determine and make the requested changes, if they can be made.

6. It is noted that the applicant did not challenge the Official Notice(s) cited in the First Office Action therefore those statements as presented are herein after prior art. Specifically it has been established that it was old and well known in the art at the time of the invention:

- the utilization of design patterns;

- that there exists a plurality of methods for comparing objects;

- enabling objects to determine (recognize), record (capture/track) and signal if any attributes have changed;

- that there exists a plurality of means for creating new and/or modified version of existing objects based on existing objects; and

- EDI's support for order and change order processes.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> **(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.**

2. Claims 1-12, 14-27 and 29-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over QAD Inc.'s MFG/PRO eB and eQ Order Management solutions as evidenced by QAD.com: Application Datasheets, Sales and Distribution, and Product pages, QAD Storefront Informational White Paper, A solution space approach white paper and Collaborative Applications Power B2B Transactions (Manufacturing Systems supplement) in view of Orr et al., U.S. Patent No. 5,191,534.

3. Regarding Claims 1, 15, 26, 30 and 33 QAD, Inc. teaches a collaborative and flexible order management (processing) system (Collaborative Applications Power B2B Transactions, Paragraphs 2-3, Page 1; Last 4 Paragraphs, Page 5; Paragraph 3, Page 7).

QAD, Inc. teaches an order management system and method as part of its extensive suite of e-business systems wherein the order management system is capable of managing sales/purchase orders and changes to orders of items offered for sale (Quad.com: Application Data Sheets Collaboration Order Management, Sales & Distribution; Pages 1 and 3-4; "...automates and streamlines the customer buying

process..."; Page 3, Paragraph 1). More generally QAD teaches a comprehensive and robust suite of products that enable e-business management (business-to-business, business-consumer, MFG/PRO, order management, etc.; QAD.com: Pages 1-4; Collaborative Applications Power B2B Transactions: "Buy-Side", "Sell-Side", "What to look for in Internet Order Management"; Page 1, Paragraph 3; Pages 4-5).

QAD, Inc. further teaches that its e-business systems utilizes a plurality of well known architectures, design elements, technologies and standards (e.g. object-oriented, N-tier, EDI, XML, etc.; Collaborative Applications Power B2B Transactions: "QAD eQ uses Java, XML and IBM's Websphere suite."; Page 5, Paragraph 2; "In QAD's case, it addressed the problem using the same object-oriented technology that allows for open integration..."; Page 3, Paragraph 3).

QAD, Inc. further teaches that the order management system and method utilizes Electronic Data Interchange (QAD.com Application Data Sheet – Sales and Distribution; Page 1, EDI support) wherein it is old and well known in the art that Electronic Data Interchange (EDI) provides for the standardized sharing of the data/information critical to the successful management and execution of order processing/management. Further it is well known that EDI includes robust support for change order processes insuring an enterprise can meet its customer's needs. More specifically EDI provides a means for communicating purchase orders and change orders; EDI standard implementations for change order processing include:

- ANSI ASC X12 (message types 850_855 - Purchase Order & Purchase Order Acknowledgment and 860_865 - Purchase Order Change & Purchase Order Change Acknowledgment) and;

- UN/EDIFACT: ORDERS (message types Purchase Order, ORDCHG - Purchase Order Change and ORDRSP - Purchase Order/Purchase Order Change Acknowledgment).

It would have been obvious to one skilled in the art at the time of the invention that the collaborative order management system as taught by QAD, Inc. would have benefited from the ability to process changes to orders as evidenced by QAD Inc.'s support for EDI. QAD, Inc.'s support of EDI standards enables the MFG/PRO eB and eQ Order Management solutions the ability to participate in industry supply/value chains wherein EDI provides well-known industry standards and practices for processing orders and change orders.

QAD, Inc. is silent on the process or steps utilized by their eB and eQ Order Management solutions for processing changes to orders.

Orr et al. teach a method and system for processing changes to orders (engineering changes, EC, manufacturing engineering changes, MEC) of items offered for sale (manufactured items being inherently produced so as to be offered for direct

and/or indirect sale; Abstract; Column 1, Lines 56-68; Column 2, Lines 1-58; Figure 3). Orr et al. teach that orders and change orders:

- are complex/composite objects (engineering change object, affected items, master item; Column 2, Lines 3-22);

- the comparison of change orders to determine affected items (impact of the changes to the order/design, differences, new values, attributes, etc.; engineering change object (EC); "The EC object contains administrative information about an engineering change, such as the designer making the change and the reasons for the change."; Column 2, Lines 14-17; Column 5, Lines 10-29; Figure 1, Element 35; Figures 5-7);

- the receipt, analysis and subsequent recording/tracking of those differences (e.g. tracking and storing previous versions of the order, affected items; "Action code column indicates the reason a master item is being changed. Typical reasons include new part introduction, changing an existing part...", Column 5, Lines 13-16; Column 7, Lines 26-61; Figures 2 and 5-8d); and

- the triggering of additional logic based on those differences ("The processing logic of the Engineering Control Manager is activated when a user selects an affected item..." Column 7, Lines 28-30; Engineering Change Control Manager, Figure 1, Element 35; Column 2, Lines 2-38; Column 7, Lines 26-40).

More specifically Orr et al. teaches the steps involved in (method for) processing changes orders (engineering changes) comprises the steps of:

- initiating (receiving) a request to change an existing order (Column 6, Lines 58-68; Figure 3);

- generating (capturing) a change order (requested change), the change order containing the changes to the order and the existing order (engineering changes database; Column 2, Lines 1-8 and 22-47; Figure 1, Elements 14 and 35);

- analyzing (comparing) the change request (order) and the existing (original) order (affected item; "After adding an affected item design engineer makes the necessary changes in the item data... and decides that the changes should be implemented."; Column 2, Lines 38-41; Column 1 Lines 21-24; Column 2, Lines 23-58; Column 4, Lines 56-68; Column 7, Lines 39-48; Figure 8B, Table 14; Claim 1);

- determining if any attributes (parameters, objects, affected items; "find all affected items..."; Column 9, Line 56) related to the change order are changed (impacted) based on the change order, and if the attributes are changed, then supplementing (recording, capturing) the change order result to indicate (show, record, signal) the differences (previous versions, changes necessary to be made, etc.; "An affected item object is created whenever an EC or MEC affects at item."; Column 4, Lines 57-58) between those attributes (Column 4, Lines 49-54; Column 5, 10-26; Column 7, Lines 28-31; Figures 1, 5 and 7); and

- providing the result of the change order and existing order comparison to one or more individuals or systems such that the recipient can distinguish the differences between the change order and the existing order (propagating the released changes

into production; Column 1, Lines 40-47; Column 2, Lines 50-58; Column 3, Lines 20-24; Column 11, Lines 38-39; Figure 5; Claim 1).

It would have been obvious to one skilled in the art at the time of the invention that the collaborative order management system, including the system's support for change order processing, as taught by QAD, Inc. would have benefited from the steps for processing changes to orders as taught by Orr et al. thereby providing a structured method for controlling, monitoring and integrating change orders (Abstract, Lines 1-3) in an enterprise, capturing change order history/versions (Column 7, Lines 46-54), and insuring items affected by changes to the order are properly understood and communicated (Column 1, Lines 40-54).

4. Regarding Claims 2 and 16 QAD, Inc. teaches an object-oriented order management system which utilizes common object-oriented design patterns, tools, architectures and technologies including but not limited to eXtensible Markup Language (XML), Java and Enterprise Java Beans (Collaborative Applications Power B2B Transactions; Page 2, Paragraph 6; Page 3, Paragraph 2; Page 5, Paragraph 1).

QAD, Inc. is silent on the process or steps utilized by their eB and eQ Order Management solutions for generating changes orders.

Orr et al. teaches a change order management system and method that:

- utilizes an object oriented architecture ("..object oriented approach.."; Column 2, Lines 3-6);

- represents a change order as a composite object (Engineering Change, EC, and Master Engineering Change (MEC); Affected Item (AI); Column 2, Lines 5-38; Figures 2, 8a and 8d) and;

- the step of generating a change order containing the changes to the existing order further comprises the steps of:

- copying the existing order (Column 7, Lines 55-58; Column 8, Lines 16-18; Figure 5, Elements 530, 550, 590; Figure 7) and;

- replacing values of any attributes (affected items) in the change order with the new values for those attributes (Column 2, Lines 38-43; Column 7, Lines 45-62).

It would have been obvious to one skilled in the art at the time of the invention that the object-oriented collaborative order management system as taught by QAD, Inc. would have benefited from the steps for processing changes to orders, including generating a change order comprising of the steps discussed above, as taught by Orr et al. thereby providing a structured method for controlling, monitoring and integrating change orders (Abstract, Lines 1-3) in an enterprise, capturing change order history/versions (Column 7, Lines 46-54), and insuring items affected by changes to the order are properly understood and communicated (Column 1, Lines 40-54; Column).

Official notice is taken that it is old and very well known in software engineering:

- that objects are defined as a data structure together with a collection of functions (methods) that act on, or refer to, that data structure;

- composite objects are a common design pattern and are defined as objects that contain other objects. A common example is that a drawing object may be composed of graphic primitives/objects, such as lines, circles, rectangles, text, and the like. The composite design pattern is used so that one can manipulate composite objects exactly in the same manner as one manipulates primitive objects.

- there exists a plurality of means for creating new (modified version of existing objects) objects based on existing objects including but not limited to the use of deep copy, constructs a new compound object and then, recursively, inserts copies of the nested objects found in the original compound object into the new compound object, and shallow copy, constructs a new compound object and then inserts the same objects into in new object that are contained the original contains. These copy-then-modify techniques insure that only the minimum amount of time is spent in creating a modified version of the original object (by only changing the modified values, instead of copying the values one at a time) and insure that any complex relationships or attributes associated with the original object are carried over to the new modified version of the object.

It would have been obvious that both the object oriented systems and methods as taught by QAD, Inc. and Orr et al would have benefited from and used a plurality of

object-oriented techniques and/or design patterns for generating change orders further wherein the modified change order is created by copying the original order object and then modifying only the order object attributes (values) which have been modified thereby insuring that the creation a change order object is conducted in an efficient manner.

5. Regarding Claims 3 and 17 QAD, Inc. teaches an object-oriented collaborative order management system as discussed above, including the ability to place a hold on order (Automatic or manual hold; QAD.com Application Data Sheet – Sales and Distribution; Page 1).

QAD, Inc. is silent on the process or steps utilized by their eB and eQ Order Management solutions for receiving changes orders.

Orr et al. teaches receiving a change to an existing change order further comprises the steps of:

- receiving identification of an existing order to be changed (EC unique identifier, Column 2, Lines 11-21; Column 10, Lines 21-35 and 65-68);

- receiving notification (signal) indicating the new value(s) for the requested changes to the order (affected item; Column 2, Lines 25-37 and 53-57; Claim 7) and;

- wherein the step generating a change order based on the existing order comprises:

- for each object in the existing order for which there is an indicated change performing the following steps:

i) copying the original order object to create a new change order object (Column 7, Lines 55-58) and as discussed above and;

ii) modifying the change order object by assigning the values of the requested changes (Column 2, Lines 38-43) and as discussed above.

It would have been obvious to one skilled in the art at the time of the invention that the collaborative order management system, including the support to place a hold on an order at any point in time, as taught by QAD, Inc. would have benefited from the steps for receiving and processing a change order as taught by Orr et al. thereby providing a structured method for controlling, monitoring and integrating change orders (Abstract, Lines 1-3) in an enterprise.

Official notice is taken that it is old and very well known in software engineering:

- that there exists a plurality of means for creating new objects based on existing objects as discussed above;

- the use of program control structures (for, while, if-then-else, etc.) to iteratively/recursively process a set of objects/information efficiently;

- that it is useful to have objects recognize and signal when changes have been made to its attributes. Such changes being recognized through the use of simple logic; comparing the new value to existing value and if the values are not the same setting a

change attribute/flag and/or sending a signal (message or method call or any of a plurality of signal means). Further it is standard software engineering practive to include this comparison logic in the get/set methods of an object, that exist to provide an indirect external interface to the object's internal attributes and;

- the use of an observer design pattern wherein the pattern defines a one-to-many dependency between a subject object and any number of observer objects so that when the subject object changes state (attribute values), all of the associated observer objects are notified and updated automatically. The observer design pattern is typically implemented so that a system remains decoupled thereby allowing the use/reuse of subject and observer objects independently.

It would have been obvious to one skilled in the art at the time of the invention that the collaborative order management system as taught by QAD, Inc. and steps for receiving and processing a change order as taught by Orr et al., would have benefited from and used a plurality of common and very well-known software engineering techniques and design patterns including: providing a signal/flag (through the implementation of the observer design pattern or get/set methods) to indicate only the modified attributes of a change order thereby providing a means for efficiently comparing order objects and the use of control structures (for repetition) to iteratively/recursively process the composite change order objects thereby providing an efficient means for processing a plurality of complex change order requests.

6. Regarding Claim 4, 8, 18 and 22 QAD, Inc. is silent on the process or steps utilized by their eB and eQ Order Management solutions for comparing changes orders.

Orr et al. teaches the comparison of change order requests as discussed above and further comprising the steps of (capturing change order history/versions; Column 2, Lines 1-17 and 38-43; Column 4, Lines 49-53; Column 5, Lines 12-29; Column 7, Lines 46-54; Figures 5, 7, 8a and 8d):

- for each object in the existing order for which there is an indicated change (affected item) generating a change order results (engineering changes, new version of design/order) that identifies (Column 2, Lines 38-42 and 48-52; Column 10, Lines 41-50; Figures 2, 5 and 7):

i) the new value of the attribute and

ii) the existing value of the attribute.

It would have been obvious to one skilled in the art at the time of the invention that the collaborative order management system as taught by QAD, Inc. would have benefited from the steps for analyzing (comparing) a change order as taught by Orr et al. thereby providing a structured method for controlling, monitoring and integrating change orders (Abstract, Lines 1-3) in an enterprise, capturing change order history/versions (Column 7, Lines 46-54), and insuring items affected by changes to the order are properly understood and communicated (Column 1, Lines 40-54; Column).

Official notice is taken that it is well known in the art of software engineering that when modifying an item (object, attribute, etc.) it is common software engineering practice to provide a means for capturing the evolution of an object or transaction, more specifically to capture snapshots of the item before and after modifications are made or transactions take place. The snapshot process can be performed at the object level by simply storing the before and after values or even a full copy of the previous (unmodified) object thereby providing a simple mechanism for rolling back unwanted changes/modifications.

It would have been obvious to one skilled in the art at the time of the invention that the object-oriented collaborative order management system as taught by QAD, Inc. would have compared the changes requested to the existing order, iteratively comparing each of the requested changes and for each requested change capturing the new and existing values thereby providing an easy way to rollback any changes request which are no longer desired or applicable.

7. Regarding Claims 5, 6, 19 and 20 QAD, Inc. is silent on the method and timing associated with comparing a change order to the existing order.

Orr et al. teaches the steps for comparing order objects as discussed above.

Orr et al. is silent on the timing associated with the comparison of a change order.

Official notice is taken that it is well known and an accepted practice that when performing a series of operations whose ultimate goal is to produce a result (document, numerical value, method call, etc...) to either generate that result concurrent with the performance of each iterative step or after all the steps have been completed; the decision of when to generate the result being an obvious design choice.

For example one might process a file containing a plurality of purchase orders wherein each order is represented as a new line of text each line containing the pertinent order information. One of the steps being performed iteratively on the file would be to parse out each of the individual orders so they can be fulfilled while concurrently generating a report showing real-time (intermediate) inventories for the items being ordered thereby allowing another system or person to monitor real-time inventory levels. However one could just as easily generate the results after the file processing has been completed choosing to display only the final inventory levels.

It would have been obvious to one skilled in the art at the time of the invention that the collaborative order management system as taught by QAD, Inc. and in view of the teachings of Orr et al. could have generated the change order results concurrent with the step of comparing the change order to the existing order (providing real-time reporting of each change order) or after comparing the change order to the existing

order had (not wasting processing time on displaying intermediate results) the choice of when to generate the change order result being an obvious design choice.

8. Regarding Claims 7 and 21 QAD, Inc. does not teach the specific structure/design of the order objects used in the MFG/PRO eB and eQ order management system.

Orr et al. teaches an object oriented system and method for controlling, monitoring and integrating change orders wherein change orders are modeled as composite objects as discussed above.

It would have been obvious to one skilled in the art at the time of the invention that the object-oriented collaborative order management system as taught by QAD, Inc. would have benefited from would have benefited from the representation of change orders as composite objects as part of a change management system in view of the teachings of Orr et al. thereby providing a structured method for controlling, monitoring and integrating change orders (Abstract, Lines 1-3) in an enterprise, capturing change order history/versions (Column 7, Lines 46-54), and insuring items affected by changes to the order are properly understood and communicated (Column 1, Lines 40-54; Column).

Further it would have been obvious to one skilled in the art at the time of the invention that the use of well-known and accepted software engineering design

patterns, specifically the representation of complex objects (existing or change orders) as composite objects would have provided a means for simplifying the complexities of orders consisting of a plurality of affected items wherein each affected items further consists of additional affected items or dependencies or parts (each item being represented by a primitive or composite object as required).

9. Regarding Claims 9, 10 and 23-25 QAD, Inc. teaches the use of EDI as part of their order management system wherein EDI provides a means for exchanging a wide variety of data, including but not limited to purchase orders and changes to purchase orders as discussed above. QAD, Inc. further teaches the use of XML as discussed above.

QAD, Inc. is silent on the method for comparing a change order to the existing order, the generation of a change order result or the subsequent format of a change order result.

Orr et al. teaches a system and method for controlling, monitoring and integrating change orders as discussed above.

Orr et al. is silent on the format of a change order result.

Official notice is taken that it is old and well known in the art that EDI provides for the inter-organizational electronic exchange of business documents in a structured, machine-process able format. EDI standards permit direct computer-to-computer exchange of formatted business transactions between business partners and makes it possible for organizations to generate, receive and process large volumes of information, swiftly and with limited human intervention. EDI provides a "language" specially designed for the processing, definition and presentation of text (markup language).

Further it is well known that at either end of an EDI transaction is a translation component which converts the standard EDI format for the transaction into the business specific format necessary for completion of the transaction thereby enabling companies to transact in a common "language" without having to convert all their existing legacy systems to the common language thereby saving time, effort and money.

It would have been obvious to one skilled in the art at the time of the invention that the collaborative order management system as taught by QAD, Inc. and further in view of the change order processing method as taught by Orr et al. would have benefited from generating the change order result in any of a plurality of formats including EDI messages and XML documents in order to facilitate the system's ability to participate in a supply chain through the utilization of well-known document formats.

10. Regarding Claims 11-12, 27, 29, 31-32 and 34 QAD, Inc. does not expressly teach the comparison of order objects.

Orr et al. teaches the analysis (comparison) of change orders objects as part of a change order processing system as discussed above. More specifically Orr et al. teach a method and system for processing changes to orders (engineering changes, EC, manufacturing engineering changes, MEC) of items offered for sale (manufactured items being inherently produced so as to be offered for direct or indirect sale; Abstract; Column 1, Lines 56-68; Column 2, Lines 1-58; Figure 3).

Orr et al. further teach a method and system for comparing order objects in an order processing system comprising ("...method for controlling and monitoring the progress of design changes from inception at a design center through implementation at manufacturing locations."; Column 1, Lines 68; Column 2, Lines 1-2):

- receiving a new value for an existing attribute (Column 2, Lines 22-48);

- copying the existing order to a change order such that the change order includes an order (peer object) corresponding to the existing order (peer object; (affected item, engineering change object; manufacturing engineering change object; Column 2, Lines 3-21; Column 7, Lines 55-58; Column 8, Lines 16-18; Figure 5, Elements 530, 550, 590; Figure 7);

- assigning the new value to the parameter change order (peer attribute of the peer object; a request to change an existing order; Column 2, Lines 22-47; Column 6, Lines 58-68; Figure 3);

- comparing the existing order (peer object) to a change order to produce a change order result indicating the differences between the existing (original, old) attribute and the new attribute in the change order (impact of the changes to the order/design, differences, new values, attributes, etc.; engineering change object (EC); "The EC object contains administrative information about an engineering change, such as the designer making the change and the reasons for the change."; Column 2, Lines 14-17; Column 5, Lines 10-29; Figure 1, Element 35; Figures 5-7);

- determining if any attributes (parameters, objects, affected items; "find all affected items..."; Column 9, Line 56) related to the change order are changed (impacted) based on the change order, and if the attributes are changed, then supplementing (recording, capturing) the change order result to indicate (show, record, signal) the differences (previous versions, changes necessary to be made, etc.; "An affected item object is created whenever an EC or MEC affects at item."; Column 4, Lines 57-58) between those attributes (Column 4, Lines 49-54; Column 5, 10-26; Column 7, Lines 28-31; Figures 1, 5 and 7); and

- providing the change order result to at least one recipient (propagating the released changes into production; Column 1, Lines 40-47; Column 2, Lines 50-58; Column 3, Lines 20-24; Column 11, Lines 38-39; Figure 5; Claim 1).

Official notice is taken that there exists a plurality of methods for comparing objects (invoking comparison logic, generating results and indicating differences from the comparison), copying existing objects and assigning new values to object attributes

are all old and well known in the art of software engineering as discussed above. Further it is well known in the art that any comparison of two or more objects would have necessitated a means for identifying the objects, which are to be compared prior to any comparison-taking place.

It would have been obvious to one skilled in the art at the time of the invention that the collaborative order management system as taught by QAD, Inc. and further in view of the change order processing method as taught by Orr et al. would have benefited from and used a plurality of well-known software engineering techniques and design patterns to facilitate the comparison of order objects.

Regarding Claim 14, claim 14 recites similar limitations to Claims 1, 2 and 11-12 and is therefore rejected using the same art and rationale as applied in the rejection of Claims 1, 2 and 11-12.

Examiner Note

Examiner has cited particular sections, pages, and paragraphs or figures in the references applied to the claims for the convenience of the applicant. Although the specific citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant, in preparing the responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the examiner.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Scott L. Jarrett whose telephone number is (703) 306-5679. The examiner can normally be reached on Monday-Friday, 8:00AM - 5:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hafiz Tariq can be reached on (703) 305-9643. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

SJ
4/1/2005

TARIQ R. HAFIZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 3600